First Hit Previous Doc Next Doc Go to Doc#
Generate Collection Print

L4: Entry 1 of 3 File: PGPB Mar 25, 2004

DOCUMENT-IDENTIFIER: US 20040059822 A1
TITLE: Network block services for client access of network-attached data storage in an IP network

Summary of Invention Paragraph:
[0008] One proposed way of addressing the problem of providing block services over an IP network is to traverse the IP stack and attempt to convert file formats to block formats, and then block formats back to file formats, in order to attach storage directly to the IP network. However, this has been seen as removing the simplicity of the network-attached solution. Therefore, IT organizations have been advised to examine network-attached solutions for file services and channel-attached solutions (e.g., Ultra SCSI, Fibre Channel) for block services, while understanding the benefits and limitations of each storage architecture. See Sean Derrington, "NAS and SAN Storage: Separate but Equal?--Part 2," File 838, Server Infrastructure Strategies, Meta Group Inc., 208 Harbor Drive, Stamford, Conn. 06912-0061, Jun. 12, 2000.

Detail Description Paragraph:
[0040] These fields include an opcode field (OpCode), a packet data unit length field (PduLen), a packet identifier field (Pid), a status field (Status), two reserved fields (Reserve1 and Reserve2), a volume ID field (VolID), an offset field (Offset), and a data length field (DataLen). The OpCode, PduLen, Status, Offset and DataLen fields in the packet header are all represented as network byte order (i.e. big endian). All bits not defined should be set to zero, and all reserve fields should be set to zero as well.

Detail Description Paragraph:
[0079] FIG. 13 shows in greater detail functional blocks in the network disk client 101 and the network block server 102 in the system of FIG. 11. FIG. 13, for example, shows that the network disk client 101 may be programmed with a plurality of application programs 131, 132. The first application 131 accesses a file system 133 layered over a volume interface 134. The volume interface 134 accesses a pseudo-disk client driver 130. However, the second application 132 directly accesses the pseudo-disk client driver 130. For example, the second application 132 accesses "raw disk" and keeps track of the logical block addresses where data is to be read or written.

Detail Description Paragraph:
[0105] The second set is a configuration command: ND_CONFIG. This is only used when the driver has been opened with both O_NDELAY and O_EXCL set. This command passes a variable length data structure consisting of a character array of 16 bytes and an array of integers. The byte array is treated as a string and is assigned to the device property vid. The array of ints is assigned to the property ipaddr. No command is provided to read the configuration information which can be readily had by prtconf(1m) or through the devinfo(7) device.

Previous Doc Next Doc Go to Doc#

Hit List

First Hit Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 4 of 4 returned.

1. Document ID: US 20050216896 A1

L10: Entry 1 of 4 File: PGPB Sep 29, 2005

PGPUB-DOCUMENT-NUMBER: 20050216896
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20050216896 A1

TITLE: Data communication via tanslation map exchange

PUBLICATION-DATE: September 29, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Doleh, Almed	Sachse	TX	US

US-CL-CURRENT: 717/136

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. Desc | Image

2. Document ID: JP 03036643 A

L10: Entry 2 of 4 File: JPAB Feb 18, 1991

PUB-NO: JP403036643A
DOCUMENT-IDENTIFIER: JP 03036643 A
TITLE: DATA PROCESSING METHOD

PUBN-DATE: February 18, 1991

INVENTOR-INFORMATION:

NAME	COUNTRY
ABE, MICHIO	

INT-CL (IPC): G06F 12/04; G06F 12/10

ABSTRACT:

PURPOSE: To process data in various address expressing forms by changing the contents of a conversion table and to prevent the overhead of preprocessing by providing an address converter, data aliner and instruction execution unit controller.

CONSTITUTION: An execution unit 4 accesses an address conversion table 9 of a main storage device and an address is fetched to a table 11 in a CPU 1, and converted by a physical address in the table. Simultaneously, an address expressing form designating bit 10 is fetched to a

copy part 12 as well and it is detected whether the data to be accessed are big Indian or little Indian. Afterwards, a main memory 2 is accessed by the address converted to the physical address. An aliner 3 puts the data in order to data arrangement to be used in an internal part according to a signal 15 showing the address expressing form from the copy part 12 and the data are delivered to the execu tion unit 4 and processed. At the time of writing, operation is made reverse. Thus, when information mixing the data in the various address expressing forms are processed, the overhead of the preprocessing can be prevented.

3. Document ID: DE 4312250 B4, DE 4312250 A1, US 5408664 A, JP 07234781 A, US 5524245 A, JP 3186905 B2

L10: Entry 3 of 4 File: DWPI Aug 25, 2005

DERWENT-ACC-NO: 1994-000815
DERWENT-WEEK: 200556

TITLE: Bi-ended firmware system for booting computer - has CPU configured to operate in first or second byte sequence operating mode by byte sequence specifying device

INVENTOR: RODRIGUEZ, R; ZARRIN, S S ; RODRIQUEZ, R

PRIORITY-DATA: 1992US-0901910 (June 19, 1992), 1995US-0378844 (January 26, 1995)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
DE 4312250 B4	August 25, 2005		000	G06F009/445
DE 4312250 A1	December 23, 1993		012	G06F009/445
US 5408664 A	April 18, 1995		012	G06F009/06
JP 07234781 A	September 5, 1995		009	G06F009/06
US 5524245 A	June 4, 1996		010	G06F009/06
JP 3186905 B2	July 11, 2001		010	G06F009/445

INT-CL (IPC): G06F 1/00; G06F 9/06; G06F 9/44; G06F 9/445; G06F 13/14

ABSTRACTED-PUB-NO: DE 4312250A
BASIC-ABSTRACT:

The firmware system includes a CPU (15) which can be configured to operate in both a first and a second byte sequence-operating mode dependent upon the state of a byte sequence specifying device.

When the CPU is reset, a second information set to the start address is recorded. The byte sequence specifying device configures the CPU to operate in the byte sequence operating mode indicated by the data element and the second programme is run. The CPU configuration is consistent with the byte sequence of the second information set.

ADVANTAGE - Computer can automatically change byte sequence numbering under full software control, with byte sequence able to be changed during system boot transparent to end user.

ABSTRACTED-PUB-NO:

US 5408664A EQUIVALENT-ABSTRACTS:

The computer system includes a device for specifying a desired one of the first and second byte-order modes, e.g. for specifying having a default condition at power-on that specifies the first byte-order mode. A CPU is configurable on reset for operation in either of the first and second byte-order modes in response to the specifying device.

A first non-volatile memory stores a first set of information including a first program, while the first set of information is characterised by the first byte order. A second non-volatile memory stores a second set of information including a second program and a data item.

USE/ADVANTAGE - For configuring computer systems for different byte orders. Allows end user to install big-indian of little indian byte ordered software under software control by effecting firmware updating without physical adjustment of hardware.

US 5524245A

A method of booting a computer system for operation in a selected one of first and second byte-order modes, the first byte-order mode for processing data characterized by a first byte order, the second byte-order mode for processing data characterized by a second byte order, comprising the steps of:

providing a CPU that is configurable for operation in either of the first and second byte-order modes;

providing a first program, stored in a first non-volatile memory, characterized by the first byte order;

providing a second program, stored in a second non-volatile memory, operable to boot an operating system;

configuring the CPU for operation in the first byte-order mode at power on;

executing the first program at power-on, including the steps of

determining the byte-order mode characterized by a signature in the second program,

writing to an appropriate CPU register to enable a corresponding serial PROM after reset,

setting a memory mapping circuit to map the second non-volatile memory to a startup address on reset; and

initiating a reset; and

on reset, configuring the CPU for operation in the byte-order mode specified by the signature in the second program, and commencing execution of the second program.

4. Document ID: JP 3556955 B2, EP 470570 A, EP 470570 A3, US 5398328 A, US 5572713 A, EP 470570 B1, DE 69124437 E

L10: Entry 4 of 4 File: DWPI Aug 25, 2004

DERWENT-ACC-NO: 1992-050636

DERWENT-WEEK: 200456

TITLE: Computer operating method on big and little indian systems - forming byte addresses with 3 to reverse access and I=O operations and also switch byte order

INVENTOR: HIMMELSTEIN, M I; KILLIAN, E A ; WEBER, L B ; HIMMELSTEI, M I ; HIMELSTEIN, M I

PRIORITY-DATA: 1990US-0564923 (August 9, 1990), 1993US-0127105 (September 27, 1993), 1995US-0379710 (January 27, 1995)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 3556955 B2	August 25, 2004		009	G06F009/44
EP 470570 A	February 12, 1992		000	
EP 470570 A3	June 30, 1993		000	
US 5398328 A	March 14, 1995		008	G06F007/00
US 5572713 A	November 5, 1996		008	G06F007/00
EP 470570 B1	January 29, 1997	E	009	G06F009/355
DE 69124437 E	March 13, 1997		000	G06F009/355

INT-CL (IPC): G06F 5/00; G06F 7/00; G06F 9/34; G06F 9/35; G06F 9/355; G06F 9/44; G06F 9/455; G06F 12/04

ABSTRACTED-PUB-NO: EP 470570A
BASIC-ABSTRACT:

A computer contains a flag which indicates whether the program being run is operating upon a Little Indian or Big Indian order of bytes in words. When the flag is set, any byte address has its least significant two bits Xor'ed with 3, which has the effect of reversing the byte access order.

The conversion can be applied at run time, or a program may be converted prior to, or during its loading into the memory. Input/Output systems also provide a byte order reversal operation when the flag is set. If preformed by software the processing penalty is 2-8.8 percent.

USE/ADVANTAGE - ALlows programs from Big or Little Indian computers to operate on the same machine.
ABSTRACTED-PUB-NO:

EP 470570B EQUIVALENT-ABSTRACTS:

A method for converting a program which uses a first predefined byte order addressing method to a program which uses a second predefined byte order addressing method, the second predefined byte order being the reverse of the first predefined byte order, comprising the steps of: finding all instructions in the program which operate on bytes of data; exclusive-ORing the lower two bits of the byte address with binary three; and replacing the lower two bits of the byte address with the result of the exclusive-ORing operation.

US 5398328A

A program designed to be executed on a computer system is converted using a first memory order to a program which is executable on a computer system employing a second memory order. The second memory order is the reverse of the first memory order. All instructions in a computer executable program which operate on bytes of data are found, each of the bytes of data having a byte address, each byte address having two least significant bits.

The two least significant bits of each byte address are combined with binary three using an exclusive-OR logic function, thus generating two complementary bits for each byte address. The two least significant bits of each byte address are replaced with the two complementary bits, thus generating a new byte address for each of the bytes of data. A detector determines whether the program is designed to be executed on a computer system employing the first memory order.

US 5572713A

A method for converting a program designed to be executed on a computer system employing a first predefined memory order to a program which is executable on a computer system employing a second predefined memory order different from said first predefined memory order, the method comprising the steps of:

finding all instructions in the program which operate on bytes of data, each of said bytes of data having a byte address, each byte address having two least significant bits;

operating on the two least significant bits of each byte address using a logic function to thereby generate two complementary bits for each byte address; and

replacing the two least significant bits of the byte address with two complementary bits to thereby generate a new byte address for each of said bytes of data.

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
BIG	105565
BIGS	722
INDIAN	20968
INDIANS	673
(9 AND (INDIAN NEAR BIG)).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	4
(L9 AND (BIG NEAR INDIAN)).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	4

Display Format: - Change Format

Previous Page Next Page Go to Doc#